AIAA Infotech@Aerospace 2010
20 - 22 April 2010, Atlanta, Georgia

AIAA 2010-3428

Visual Marker Detection In The Presence Of Colored Noise for Unmanned Aerial Vehicles

Syed Irtiza Ali Shah[1], Allen D. Wu[2] and Eric Norman Johnson[3]
Georgia Institute of Technology, Atlanta, GA 30332

This paper develops a vision-based algorithm to detect a visual marker in real time and in the presence of excessive colored noise for Unmanned Aerial Vehicles. After using various image analysis techniques, including color histograms, filtering techniques and color space analyses, typical pixel-based characteristics of the visual marker were established. It was found that not only various color space based characteristics were significant, but also relationships between various channels across different color spaces were of great consequence. A block based search algorithm was then used to search for those established characteristics in real-time image data stream from a colored camera. A low cost noise and interference filter was also devised to handle excessive noise that was encountered during flight tests. The specific implementation scenario is that of detection of a Blue LED for GerogiaTech's participating aircraft into the International Aerial Robotics competition. The final algorithm that was implemented on GTAR lama aircraft, used both multiple thresholding and linear confidence level calculations and was successfully used in the competition in 2009.

Nomenclature

UAV	=	Unmanned Air Vehicle
AUVSI	=	Association for Unmanned Vehicle Systems International
GTAR	=	GeorgiaTech Aerial Robotics
YCbCr	=	luma, blue-difference and red-difference chroma color space
Lab	=	L*, a*, b* color space (CIELAB) of lightness and a & b color components
MPEG	=	Motion Pictures Expert Group
LED	=	light emitting diode
L_{th}	=	threshold level of a color component
R,G,B	=	red, green and blue color components respectively
N_R, N_G, N_B	=	number of red, green or blue pixels above L_{th} threshold level
f(x,y)	=	input image to Laplacian filter
g(x,y)	=	enhance image after Laplacian sharpening
c	=	constant indicating sign of center element of Laplacian mask
MinX	=	minimum level of any color space parameter X
Confd_X	=	confidence level for presence of blue LED based on parameter X

I. Introduction

IN this work we aim to design a low cost vision system for GeorgiaTech's GTAR (GeorgiaTech Aerial Robotics) aircraft for participation into the 5th mission of the International Aerial Robotics Competition 2009, organized by Association for Unmanned Vehicle Systems International (Ref. 1).

The competition required creating a small aerial robot capable of fully autonomous flight through a confined environment such as a nuclear power generating plant. The vehicle was first required to enter the building through a one square meter opening from a designated launch area three meters away. The vehicle was then to search for a

[1] Graduate Research Assistant, School of Aerospace Eng, 270 Ferst Drive, Atlanta GA 30332, AIAA Member.
[2] Graduate Research Assistant, School of Aerospace Eng, 270 Ferst Drive, Atlanta GA 30332, AIAA Member.
[3] Lockheed Martin Associate Professor of Avionics Integration, School of Aerospace Eng, 270 Ferst Drive, Atlanta GA 30332, AIAA Member.

target area while avoiding obstacles such as walls, columns, and furniture. Upon locating the control panel which was to be identified by various blinking lights and an audible warning tone, a picture of a specific target gauge, still or video, was to be relayed by means of a radio frequency (RF) signal with sufficient power to be received at a distance of 100 meters with a loss of 6 dB (to account for attenuation by the structure). The target gauge of interest was the only one with a non-blinking blue LED directly below it.

Hence to solve this problem of searching and detecting a visual marker such as a blue LED, we considered various color, shape and object recognition techniques in the literature, along with some block search algorithms. In order to handle the significant noise and interference that was encountered, a few noise handling techniques were also considered. However, based on lab experiments and analysis, we devised our own algorithm, which after requisite tuning based on flight tests, worked remarkably well on GeorgiaTech's UAV platform in actual flights.

A. Color / Shape recognition and detection

There has been extensive research in the area of color and shape recognition and detection in a stream of images. One of the relevant examples includes Ref. 2, in which authors propose a generic rule based color model for fire pixel detection using YCbCr color space to separate the luminance from the chrominance for effectively addressing the issue of illumination variations. A set of rules defined on the Y, Cb and Cr color components together with the developed chrominance model on the Cb-Cr color plane were used to detect the fire pixels in color images. Similarly, in Ref. 3, authors propose a vision-based street detection algorithm to be used by small autonomous aircraft in low-altitude urban surveillance. This algorithm uses Bayesian analysis to differentiate between street and background pixels. The color profile of edges on the detected street is used to represent objects with respect to their surroundings. These color profiles are used to improve street detection over time. Pixels that do not likely originate from the "true" street are excluded from the recurring Bayesian estimation in the video. Yet another relevant approach is that of Ref. 4, in which shadow detection of color aerial images based on a successive thresholding scheme has been presented by the authors. By performing the global thresholding process on a ratio map, a coarse-shadow map is constructed to classify the input color aerial image into the candidate shadow pixels and the non-shadow pixels. For each candidate shadow region, the local thresholding process is performed iteratively to extract the true shadow pixels from the candidate shadow region. Finally, for the remaining candidate shadow regions, a fine-shadow determination process is applied to identify whether each remaining candidate shadow pixel is the true shadow pixel or not. In Ref. 5 authors propose object detection by a color histogram-based fuzzy classifier with support vector learning. They use hue and saturation color space histograms to detect desired features.

In Ref. 6, road recognition based on color image edge detection has been considered. They have used 'Lab' color space (which approximates human vision,) along with Hough transforms to extract edges of the road. In Ref. 7 traffic sign shape recognition using geometric matching has been proposed. To reduce digital noise and extract the shape of each individual traffic sign, the external boundaries of traffic signs were segmented based on color information and were simplified and decomposed through discrete curve evolution, whose stop stage was determined by an arc similarity measure in tangent space. The recognition of a closed candidate shape was achieved through direct matching with templates. An optimal enclosure was generated to minimize the geometric differences between the retrieved unclosed candidate shape and templates. We in this paper after consideration of all these approaches started off with an approach that is comparable to Ref. 2.

B. Block Based Searches

In Ref. 8, block motion estimation algorithm based on a full-search array architecture has been proposed. The modified gradient-descent search (MGDS) algorithm utilizes an adaptive computation distribution mechanism to allocate computation of the employed ±1 full search array to blocks or frames of video sequences. In Ref. 9 a hierarchical design methodology for full-search block matching motion estimation has been proposed which decomposes a 2-D full search problem into several 1-D functional blocks, where the design of each block could be optimized separately. In Ref. 10 a full search successive elimination algorithm (SEA) for variable block-based motion estimation of MPEG/H.264 standard has been proposed. They apply SEA in a hierarchical manner to reduce computational complexity. We in this paper used a linear full search block based algorithm with fixed as well as variable grid sizes on the image.

C. Colored Noise Handling / Filtering

For noise handling we considered Ref. 11, which proposes color image noise removal to eliminate noisy pixels by exploiting several vector-class characteristics of multichannel pixels. This algorithm treats multichannel images as a vector class and takes both magnitude and phase angles of the pixel vectors into consideration. Another approach is that of Ref. 12, in which white and color noise cancellation using an adaptive feedback cross-coupled

line enhancer filter is proposed. This method consists of two adaptive FIR(finite impulse response) filters, where the output of each filter is fedback to the input of the other filter to form an adaptive feedback cross-coupled filter. Based on the image processing and analysis of the noise we encountered, we used a very simple thresholding noise filter which worked very effectively even for noise as high as 40% in our image data.

II. Preliminary Experiments and Analysis

With the low cost camera we intended to use on our UAV, we devised an experimental set up (shown in Fig. 1). It comprised of a color video camera (Panasonic CCTV Board Camera) with a wireless transmitter and a power supply, a receiver with VCE-Pro PCMCIA video capture card / frame grabber, various LEDs, and a laptop to process images. For this testing we used a 2.4GHz wireless link. We took images varying between 1 ft to 6 ft distances. Beyond 6 ft, the font 18 size on the target gauge, as given in the problem description[1], would have been unreadable. Hence, it was not required to attempt to detect the blue LED from beyond 6 ft distance from it. We took quite a few images of red, blue, yellow and green LEDs and images with all or no LEDs at all. We also used a blue LED separate from this experimental setup in order to do further analysis on such images. Some sample input images are shown in Fig. 2.

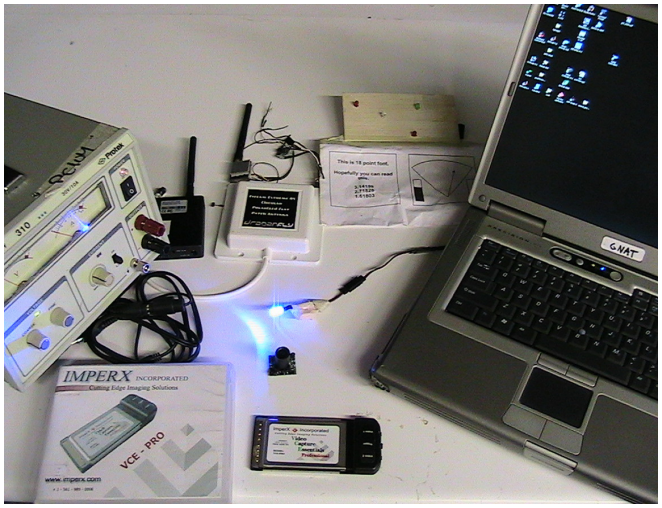


Figure 1. GTAR Lama Camera and Experimental Setup used. *Panasonic CCTV Board Camera with a transmitter and a power supply, a receiver with VCE-Pro PCMCIA video capture card, various LEDs, and a laptop for processing images.*

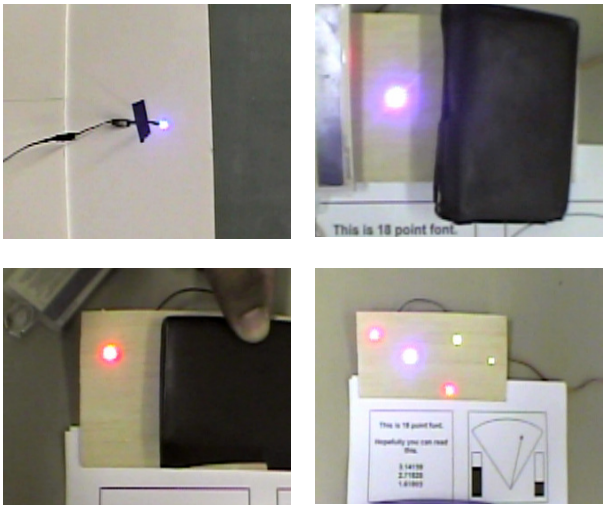


Figure 2. Some Sample Input Images. *27 such test images were used for preliminary analysis*

.

Intensity transformation functions based on information extracted from image intensity histograms play a basic role in image processing in the areas of enhancement, compression, segmentation and description [13] . We also performed Histogram Equalization, which is a technique to achieve enhancement by spreading the levels of the input image over a wider range of the intensity scale [13]. Hence the histograms were plotted both for the grayscale image and the equalized image for all test images. Results for a sample image are presented in Fig. 3. This however, did not reveal any quantifiable information to detect the blue LED, or to separate an image with a blue LED from an image without a blue LED. We then did the histogram analysis separately on red, blue and green channels of the images (Fig. 3).

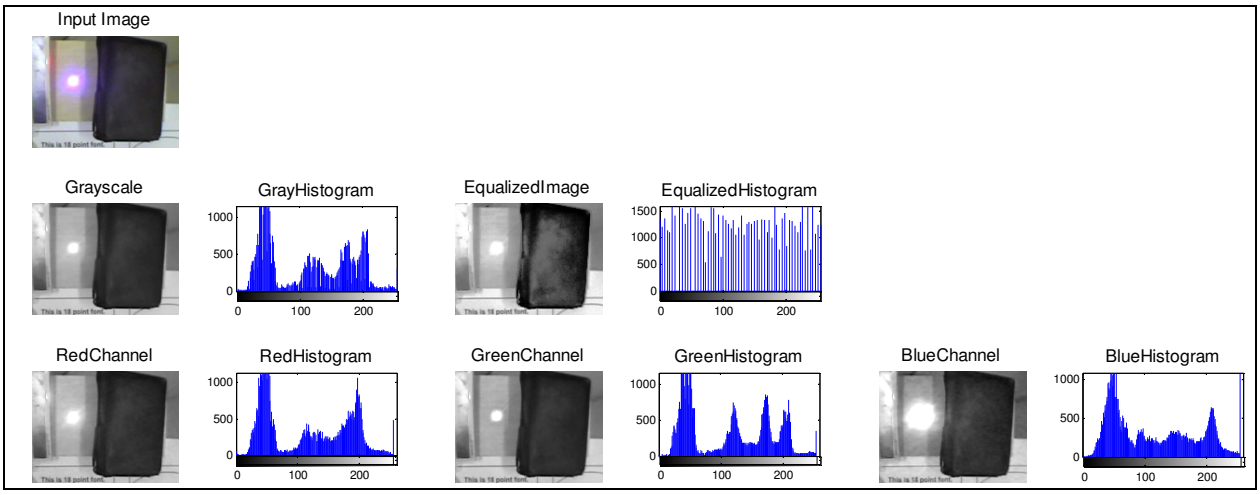


Figure 3. Grayscale Histogram with Histogram Equalization along with Histograms for Separated Red, Blue and Green Channels. *Similar analysis was done for all 27 test images.*

We also performed similar analyses on various color spaces for the test images. This included the following color spaces (refer Fig. 4 for a sample analysis on a test image):

NTSC color space: luminance-Y, hue-I and saturation-Q channels
YCbCr color space: luminance-Y, blue difference-Cb and red difference-Cr channels
HSV color space: tint-H, shade-S and tone-V channels
CMY color space: cyan, magenta and yellow channels
HSI color space: hue-H, saturation-S and intensity-I channels

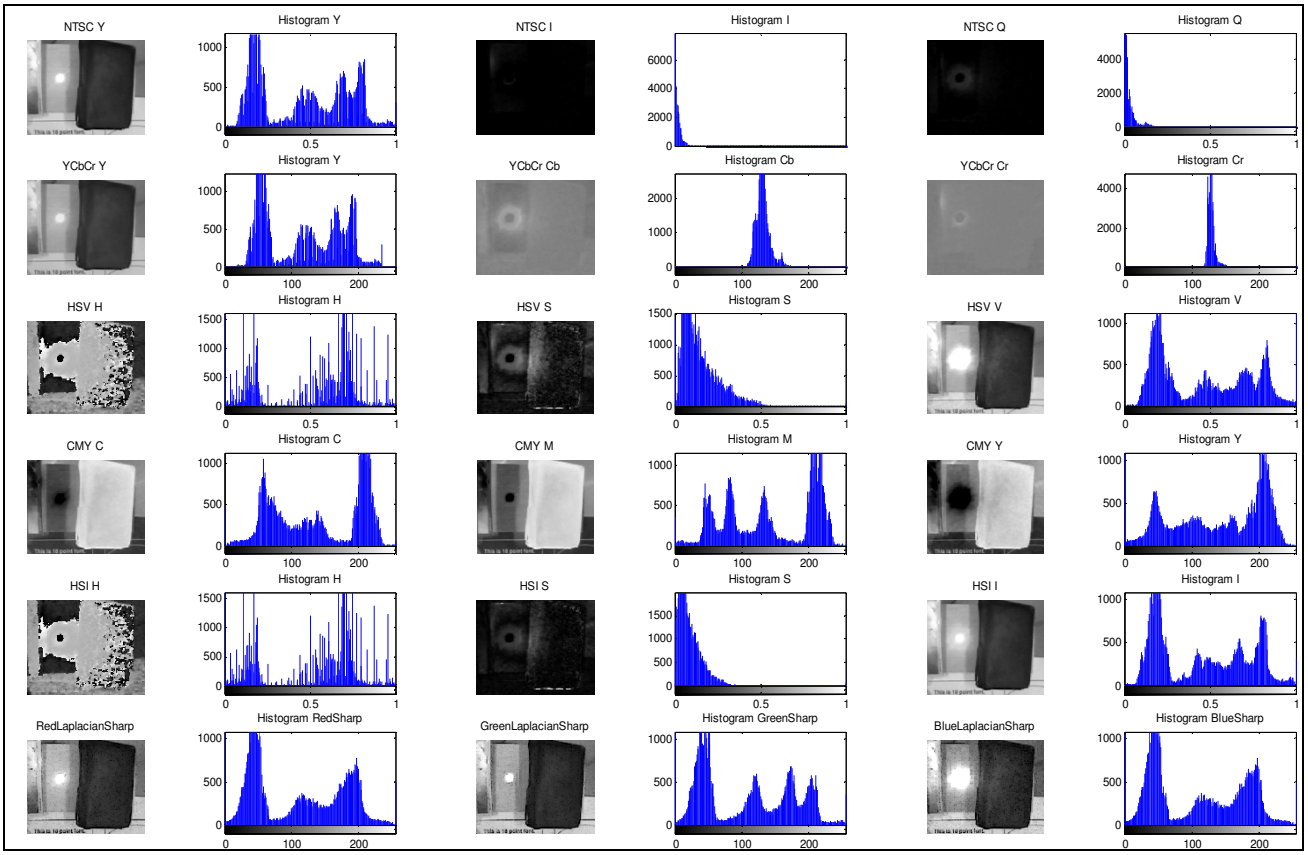


Figure 4. Histograms and Laplacian Filtering for Various Color Spaces. *Similar analysis was done for all initial 27 test images.*

After similar analysis was performed on all captured test images, we concluded that the number of pixels with high blue channel values, as compared to number of pixels with high red or green channel values, was significantly high in the images containing a blue LED, than the number of such pixels in images without a blue LED. Thus for example, if we choose a threshold level L_{th} and count the number of pixels with higher levels of blue, red and green, than this L_{th} for a test image, i.e.,

For blue level $B > L_{th}$, count all such pixels totaling to N_B,
For red level $R > L_{th}$, count all such pixels totaling to N_R,
For green level $G > L_{th}$, count all such pixels totaling to N_G,

then if

$N_B > N_R$ such that $N_B/N_R \approx 1.5$
$N_B > N_G$ such that $N_B/N_G \approx 2.0$

Then the test image contained a blue LED

(On 8-bit image pixels, where pixel values vary from 0-255 for each of three channels, the L_{th} was observed to be around 200.)

We then wrote a test Matlab code, to check this observation. It was found that this code resulted in 80% success on the chosen 27 test images. Once the code was tested against many more images, containing bright white and bright blue colors from sources other than LEDs, the code gave false positives. This was for obvious reasons of having large number of high blue channel valued or high white valued pixels in such images. Hence we concluded that we could not detect a blue LED with practical certainty, using this method.

We also attempted to use a blue filter over the lens of our camera and captured images using one layer of the filter as well as two layers. Upon analysis of such images, we could not conclude any additional facts for our application.

Besides, we also attempted image enhancement using Laplacian filtering[14] which is based on the following equation

$$g(x,y) = f(x,y) + c[\nabla^2 f(x,y)]$$

where $f(x,y)$ is the input image, $g(x,y)$ is the enhanced image and c is 1 if the center coefficient of the mask is positive, or -1 if it is negative. We used a 3x3 Laplacian sharpening mask as follows (with c=-1):

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The result on one test image is shown in Fig. 4 (last row). However, this filtering did not give us any additional information for our application. As a next step, we cropped only the LED part from the image and attempted to perform similar analysis.

III. Detailed Analysis and Proposed Block Based Search Algorithms

A. Detailed Analysis

For our detailed analysis of cropped out images of specific LED parts, the goal was to establish a generic rule based model for a blue LED as seen from our camera in an image (similar to reference 1, in which similar methodology was used to detect fire in the images). Hence we analyzed not only a blue LED, but also other LEDs images including red, green and yellow, so as to clearly differentiate a blue LED from other LEDs. We chose 59 such images with just the LED showing, including 29 images of blue LED, 8 of green, 7 of yellow and 15 of red (refer Fig. 5).

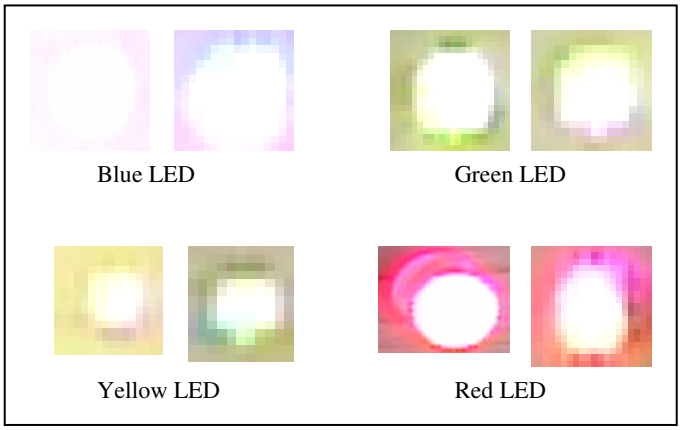
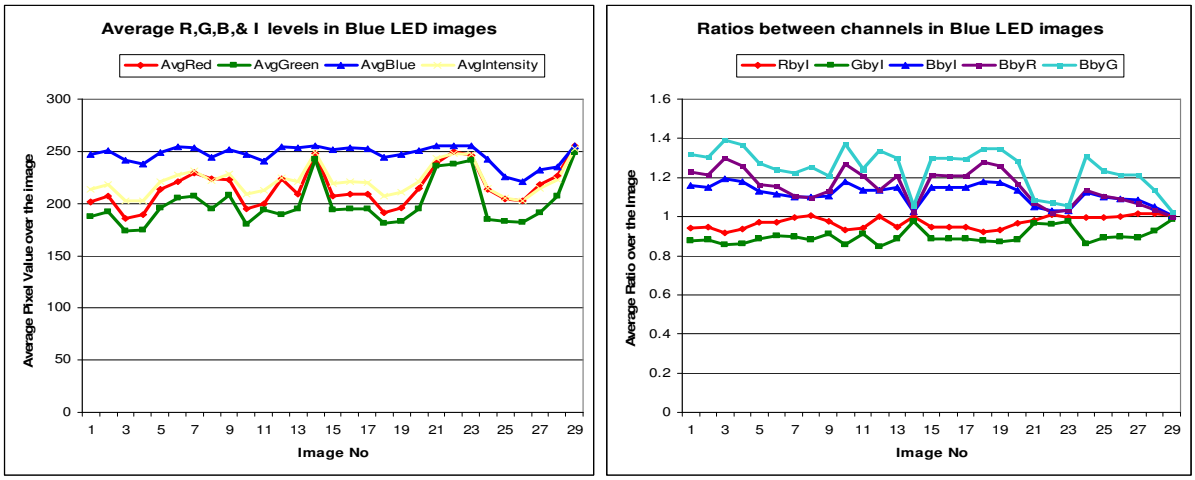


Figure 5. Cropped Images of LEDs. *59 such images were used in total, including 29 images of blue LED, 8 of green, 7 of yellow and 15 of red*

We calculated the following nine parameters for each of the cropped images,

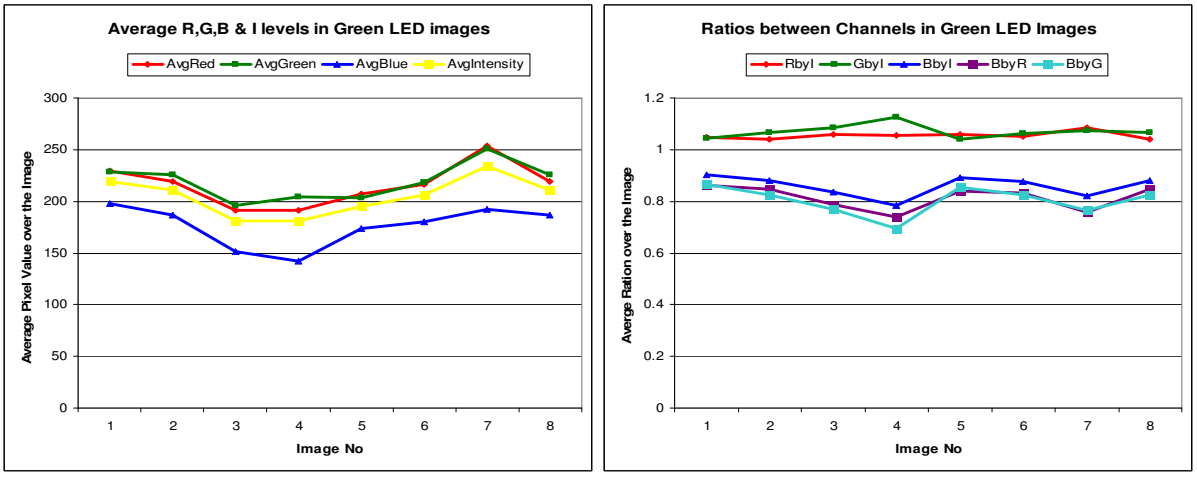
1) Average value of red level of the cropped image showing only the LED
2) Average value of blue level of that image
3) Average value of green level of that image
4) Average value of intensity of that image
5) Ratio between average red level and average intensity level of that image
6) Ratio between average green level and average intensity level of that image
7) Ratio between average blue level and average intensity level of that image
8) Ratio between average blue level and average red level of that image
9) Ratio between average blue level and average green level of that image

The results from this analysis are as shown in Figs. 6, 7, 8 & 9.



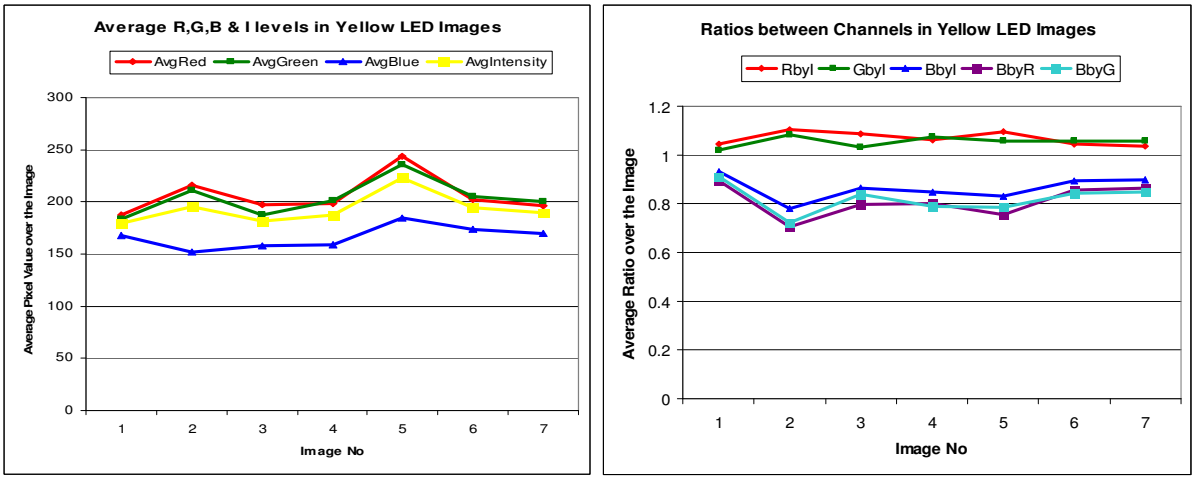
a) *b)*

Figure 6. Color Space Analysis for Blue LED Images.



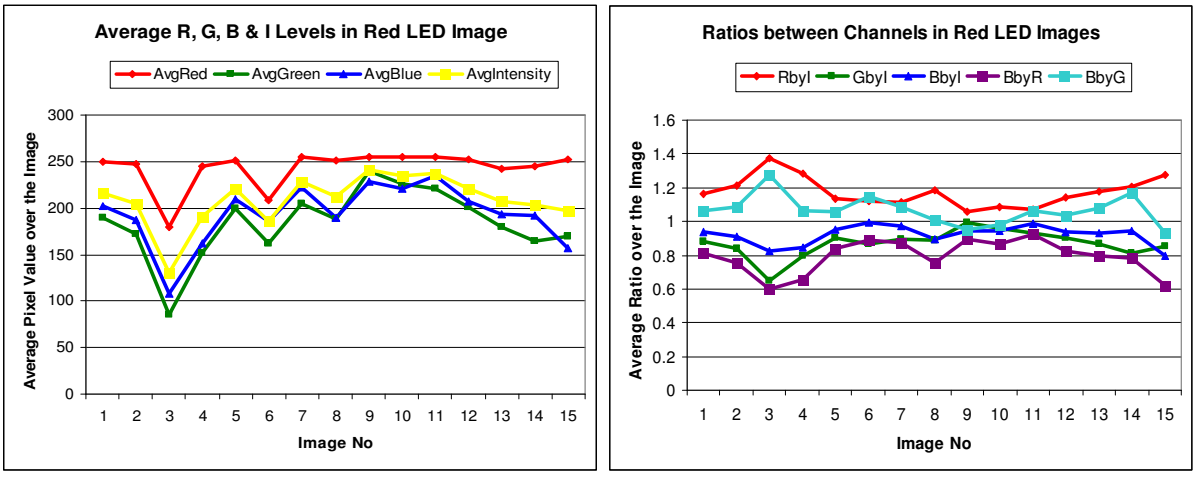
a) *b)*

Figure 7. Color Space Analysis for Green LED Images.



a) *b)*

Figure 8. Color Space Analysis for Yellow LED Images.



a) *b)*

Figure 9. Color Space Analysis for Red LED Images.

For the respective LED images, a summary of the above data analysis in terms of observed minimum, maximum and average values along with standard deviations is presented in Table 1. This gives an idea of typical characteristics of these LEDs as seen in our experimental setup.

Since we were required to detect only a blue LED, we concentrated on those characteristics of a blue LED which are could be identified as peculiar characteristics of a blue LED. Hence from the above analysis (presented in Figs. 6 to 9 and Table 1), we deduced the following

1) We can define a minimum average blue level (Min_B) in all images having a blue LED. This Min_B value was observed to be significantly higher than all other LED images. (Referring to Table 1, it is 221 for Blue LED images, as compared to 142, 152 and 108 in green, yellow and red LEDs respectively. Also refer Figs. 6 to 9).
2) We can also define a minimum intensity level (Min_I,) which also is observed to be significantly higher for blue LED images than for green, yellow and red LEDs (200 for Blue LED as compared to 179, 180 and 124 for green, yellow and red LEDs, respectively in Table 1). (This would however, be true for a white LED as well, and unless coupled with other criteria, in itself is an insufficient identification criteria for identification of a blue LED).
3) The average value of blue content in blue LED images is always significantly higher than other colors. This behavior is not observed in all other LED images and hence is identified as a peculiar characteristic of a blue LED. This value is 247 for the Blue LED images being analyzed, as opposed to 176, 166 and 193 respectively for green, yellow and red LEDs (refer Table 1).

4) Specifically comparing average blue levels with average red levels, the ratio $Min_{B/R}$ is observed to be at least 1.000 (Table 1). This value is much higher than the ratio $Min_{B/R}$ for the green, yellow and red LEDs (0.741, 0.706 and 0.602 respectively in Table 1).
5) Similarly comparing average blue levels with average green levels, the $Min_{B/G}$ value is observed to be 1.024. Again this value is significantly higher than $Min_{B/G}$ values for the green, yellow and red LEDs (0.695, 0.723 and 0.931 respectively in Table 1).
6) In the same way, comparing average blue levels with average intensity levels, the $Min_{B/I}$ value is observed to be 1.008. This value, is significantly higher than $Min_{B/I}$ values for the green, yellow and red LEDs (0.792, 0.789 and 0.816 respectively in Table 1).

TABLE 1. Summary of Color Space Analysis.

		Average Channel Values of an LED Image				Ratios between Channels				
		Avg Red	Avg Green	Avg Blue	Avg Intensity	R / I	G / I	B / I	B / R	B / G
Blue LED	Min	186	173	**221**	**200**	0.927	0.852	**1.008**	**1.000**	**1.024**
	Max	254	245	255	253	1.019	0.984	1.206	1.3	1.391
	Avg	216	200	**247**	221	0.976	0.904	1.121	1.151	1.244
	Std Dev	18.36	20.83	9.05	14.64	0.029	0.035	0.054	0.085	0.102
Green LED	Min	192	196	**142**	**179**	1.043	1.044	**0.792**	**0.741**	**0.695**
	Max	253	251	198	232	1.092	1.14	0.906	0.863	0.867
	Avg	216	219	**176**	204	1.06	1.076	0.864	0.815	0.804
	Std Dev	18.92	16.41	18.53	17.21	0.015	0.028	0.037	0.043	0.053
Yellow LED	Min	188	184	**152**	**180**	1.039	1.022	**0.789**	**0.706**	**0.723**
	Max	244	236	185	221	1.118	1.093	0.933	0.893	0.912
	Avg	206	204	**166**	192	1.072	1.06	0.868	0.811	0.82
	Std Dev	17.37	15.75	10.22	13.03	0.029	0.022	0.044	0.061	0.056
Red LED	Min	180	85	**108**	**124**	1.057	0.682	**0.816**	**0.602**	**0.931**
	Max	255	240	235	241	1.447	0.996	1.001	0.923	1.275
	Avg	243	183	**193**	207	1.189	0.879	0.932	0.792	1.066
	Std Dev	20.22	36	31.47	27.87	0.101	0.07	0.048	0.096	0.083

B. Multiple Thresholding Algorithm

Based on the findings from the image analysis presented above, the first approach that was considered was Multiple Thresholding. A block based search algorithm was used to find a block of pixels in an image which passes all the chosen thresholds. If such a is found, it is most probably a block of pixels belonging to a blue LED. Keeping in view Table 1, following fixed thresholds were chosen

Min_B = 220
Min_I = 200

The other three thresholds chosen were $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$. However, suitable values for these were found by repetitive testing with real time videos from the onboard camera.

Hence, the implementation of this algorithm required that a block of pixels pass all these five thresholds in order to be declared as the most probable one belonging to a blue LED in the image. An initial Matlab implementation of such a linear full search block matching algorithm with block size kept at 8x8 pixels in a fixed grid gave the following results on the initial sample of 27 test images

1) Choosing minimum observed values for $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$ (i.e 1.008, 1.000 and 1.024 from Table 1, respectively), the algorithm identified all 17 blue LED images out of 27 but gave 9 false positives as well, which was unacceptable.
2) Choosing minimum plus one standard deviation values for $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$ (i.e 1.008+0.054=1.062, 1.000+0.085=1.085 and 1.024+0.102=1.126 respectively, referring to Table 1), 16 blue LED images out of 17 were correctly identified, but it gave 5 false positives out of 9. This was significantly better than the ones obtained by choosing minimum values, as in the first case above.
3) Choosing minimum plus 2 times standard deviation values (i.e. 1.115, 1.169 and 1.228), correctly identified 14 blue LED images out of 17 and did not give any false positives out of the rest 9 images. This was a further improvement from the previous step 2.
4) Choosing mean values (i.e. 1.121, 1.151 and 1.244 from Table 1) also correctly identified 14 blue LED images out of 17 and did not give any false positives. The success rate in this case was the same as case 3 above.
5) Choosing minimum plus 3 times standard deviation values (i.e. 1.169, 1.254 and 1.33), identified only 9 blue LED images out of 17 but did not give any false positives. Hence these thresholds were more stringent than case 3 or 4 above.
6) Choosing maximum values for $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$ (i.e. 1.2, 1.3 and 1.39), correctly identified only 5 of the 17 total blue LED images and did not give any false positives. These were the most stringent thresholds and would only pick very bright looking blue LEDs in the images.
7) A few more iterations of Matlab code were run to find some empirical values for identifying all 17 blue LED images avoiding any false positives. These values came out to be 1.107, 1.08 and 1.228 respectively, for $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$.

However, it may be said that these results were good for only the 27 test images that were chosen initially, and for a block size of 8x8 pixels in a fixed grid on the image. These thresholds therefore, needed to be further tested not only with other block sizes and with variable grids, but more importantly, to be extensively tested in the final implementation of this algorithm with real time video from the onboard camera of our aircraft. This testing was required during both hardware in the loop and actual flight tests in order to confirm the efficacy of this proposed algorithm.

C. Linear Confidence Level Algorithm

Apart from multiple thresholding proposed above, another algorithm is also proposed, which defines a confidence level for the blue LED detection in an image. For all the five parameters considered i.e *MinB , MinI , MinB/I , MinB/R and MinB/G,* a confidence level for each of these parameters X, is defined as follows

$$Confd\_X_k = (Confd\_X_{Max} - Confd\_X_{Min}) \cdot (X_k - Min_X) / (Max_X - Min_X) + Confd\_X_{Min}$$

where subscript *k* is for the value of the parameter calculated for the block of pixels under consideration. It may be noted that this simply, is a straight line approximation of the confidence level for that parameter, as depicted in Fig. 10.

We choose minimum and maximum confidence level values ($Confd\_X_{Max}$ and $Confd\_X_{Min}$) as 0.7 and 1.0 respectively. This means that if a block of pixels crosses the minimum threshold on this parameter, we consider it to be 70% probable as being a block belonging to blue LED. Similarly if it crosses the max threshold, we are fully confident for this specific parameter that the block under consideration belongs to a blue LED. The minimum and maximum values of the five chosen parameters (Min_B , Min_I , $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$) were taken from Table 1.

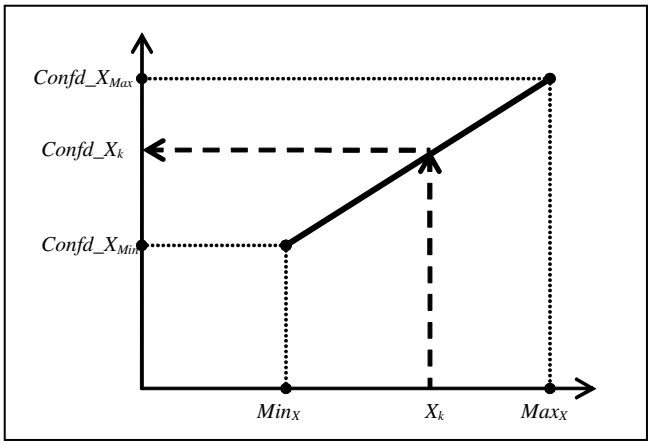


Figure 10. Linear Confidence Level Calculation for a Chosen Parameter.

After defining a confidence level for individual parameters, we defined an overall confidence level for overall certainty of the block under consideration containing a blue LED as follows

$$Confidence_{Overall} = \prod_{j=1}^{5} Confd\_X_j \qquad (j=1,..,5 \text{ for five chosen parameters})$$

An initial Matlab implementation of this linear full search block matching algorithm with block size kept at 8x8 pixels in a fixed grid over the image gave the results as shown in Fig. 11 on our initial sample of 27 test images.

It may be observed from Fig. 11 that almost all blue LED images give higher confidence levels than other LED images. Like the multiple thresholding case, it may be mentioned here again that these results were based on only the 27 test images chosen initially and for a block size of 8x8 pixels in a fixed grid on the image. This proposed algorithm therefore, also needed to be further tested not only with other block sizes with variable grids, but more importantly, to be tested in the final implementation of this algorithm with real time video from the onboard camera of our aircraft. In fact, this linear confidence level algorithm was the first algorithm that was physically implemented on our system for HITL and flight tests.

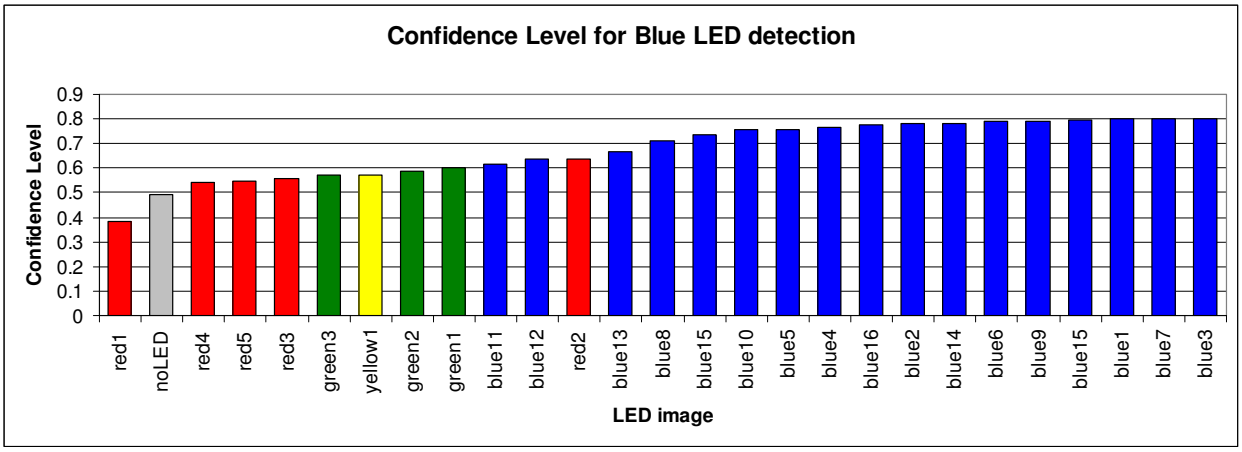


Figure 11. Overall Confidence Level Calculation on 27 Test Images Containing Different LEDs. *Almost all images containing blue LED give higher confidence levels than other LED images.*

IV. Implementation and Flight Testing

A. Hardware Overview

The aerial platform used for implementation of this research is a coaxial miniature rotorcraft platform, which inherently possesses a stick-free stability. It has a pair of counter-rotating blades, making the vehicle more compact since no tail rotor is required for yaw control. The bottom set of blades has cyclic control for maneuvering, while the upper set of blades has a Bell stabilizer (also known as a flybar) to counteract vehicle pitch and roll thus providing attitude stability. The helicopter initially selected as a base platform was the E020 Big Lama, made by E-Sky. It has a rotor diameter of 46cm and weighs approximately 410g in its stock configuration. Initial flight tests indicated that the stock vehicle has a useful payload capacity of approximately 50g which was further increased by removing the canopy and upgrading the motors. The final flight configuration, with avionics, a larger battery, brushless motors, and a protective shroud, weighed 605g with a rotor diameter of less than 1 meter.

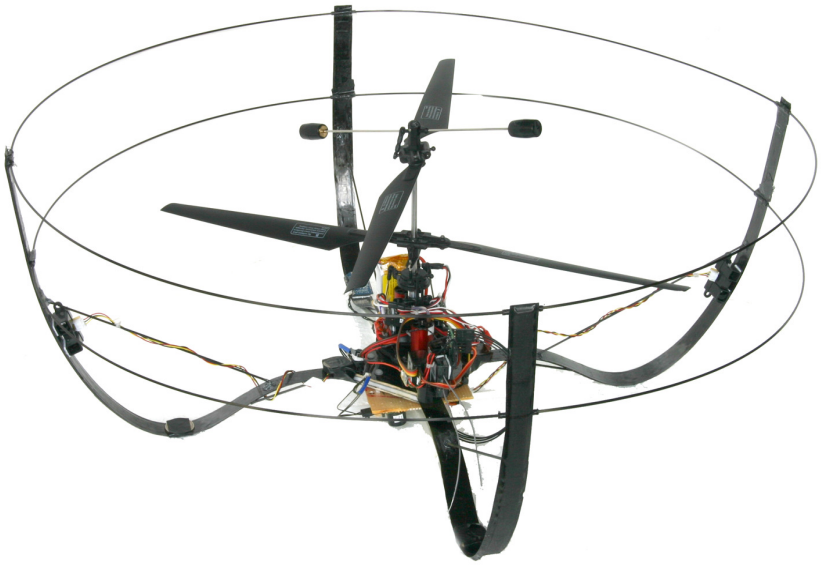


Figure 12. GTAR Lama Aircraft

Off-the-shelf range sensors were used to provide local position information. The MaxBotix LV-EZ1 sonar was chosen for measuring altitude, and the SHARP GP2Y0A02YK0F infrared sensor was selected for measuring range to walls and obstacles. Range measurements from the IR sensors were also used to estimate vehicle heading. An ATMega128 onboard microprocessor was used to read the sensors and process the data for navigation. The data from these sensors was processed and filtered appropriately according to their error characteristics for simple obstacle avoidance and wall-following behavior. The IR range sensors have analog voltage outputs which are read by onboard analog to digital input channels. The altitude sonar is read via serial port. Data is transmitted to the ground via a wireless link using an additional serial port on the ATMega128. The sonar is mounted pointing downward. Two IR sensors are placed 45cm apart looking forward for heading control, for improved obstacle avoidance during forward motion, for detection of openings such as windows or doors, and for wall-following behavior. In addition, one IR sensor each is pointed left and right to detect obstacles and oncoming walls during wall following flight. For the vision system, we used a Panasonic CCTV Board Camera with a VCE-Pro PCMCIA video capture card / frame grabber. Initially we used 2.4GHz links for all equipment, but because of excessive interference in video from all other onboard equipment, we switched the video link to 900 MHz exclusively for the vision system.

B. Onboard Software Overview [15]

The navigation algorithm comprised of a smart filtering routine to prevent large step or impulse inputs from adversely affecting the controller. A Kalman filter was used to estimate the range, and the covariance of the residuals was used to detect and ignore outliers beyond three standard deviations. For guidance, the vehicle follows the wall by flying laterally while detecting and maneuvering corners using side-looking range sensors. A separate altitude-hold controller is used to maintain a fixed altitude throughout the flight. Initial entry into the building is handled by a "Window Entry" logic. In this mode, an object detected on the forward-looking left or right IR sensors causes the lateral controller to adjust the flight path to remain centered on the window. Once the vehicle enters the competition arena, walls are detected by the left IR sensor and the vehicle enters "Left Turn" mode. In this mode, an open-loop left turn is commanded until the forward-looking IR sensors detect the wall and "Wall Follow" mode begins. In this mode, the longitudinal controller maintains a commanded distance from the wall while the heading controller maintains the desired heading with respect to the wall. The vehicle then uses an open loop lateral command to fly along the wall to the right, using the right facing sensor to detect walls and obstacles in the flight path. Once an obstacle or wall is detected by the right looking IR sensor, the different corner-turning modes are initiated. If a wall or obstacle is detected in the direction of flight, the vehicle enters "Inside Turn" mode, whereby it changes its heading to either turn the corner (for concave corners) or fly around the obstacle. This is achieved by giving an open-loop yaw command until no obstacle is seen by the right IR sensor. If one of the forward-looking IR sensors detects a step change to max range while the other sensor still reads near the estimated wall distance, a convex or outside corner has been detected. The vehicle then enters "Outside Turn" mode and it yaws to the left in

order to continue around the corner. Once an inside or outside corner has been turned, and valid range measurements are seen on the two front IR sensors, the vehicle returns to "Wall Follow" mode and continues flight until the entire indoor area is traversed.

For the control algorithm, since the coaxial rotorcraft platform inherently possesses attitude stabilization, the controller only provides servo deflections such that the vehicle is able to track a commanded direction. Although the controller has access only to the local range information, by exploiting the fact that indoor structures have walls, the controller can be designed to follow the walls. This circumvents the requirement for a global position fix. The control architecture used is a Proportional Integral Derivative (PID) design with gain scheduling applied such that the controller uses different gain values depending on where the vehicle is with respect to the wall. A Kalman filter based local velocity estimator is used in lieu of measured data. The position command is then directly linked to the actuator deflection using a PID control logic. The control action is thus achieved by using four independent control loops, i.e Altitude Hold, Heading Hold, Longitudinal Position Control and Lateral Position Control.

C. HITL tests and Flight Tests

For hardware in the loop tests, only the aircraft camera was on, and we were manually carrying the aircraft around to simulate the flight. At first, only the Linear Confidence Level algorithm (Section III-C) was implemented in Visual C++ and integrated with the aircraft software as described in Section IV-B above. Depending on these tests, necessary tuning of values was done for all five parameters: Min_B , Min_I , $Min_{B/I}$, $Min_{B/R}$ and $Min_{B/G}$ so as to get a satisfactory confidence level for detection of the blue LED and avoiding false positives from bright or light blue colored objects in the scene. Unlike the Matlab implementations until now, a variable block size with a variable grid on the image was used in this real time implementation. The grid was controlled using a variable as a 'grid step' with value varying anywhere between 1 and the block size. A grid step of 1 would mean that a block would be picked starting from every pixel in the image. Though this would increase the probability of finding the blue LED block giving a high overall confidence level, but will increase the computational load quite a bit. A 'grid step' value equal to block size would mean a fixed grid on the image. This would make the algorithm very fast as very few calculations would be required on such a fixed grid. However, it would give lesser overall confidence levels, especially for a case where a blue LED overlaps the edges of this fixed grid block under consideration. The tests revealed that if a grid step of less than half of the block size is chosen, the computational load would increase to an extent that a real-time application would not be practical. Hence, the grid step was subsequently kept at half of the block size.

This 'hardware in the loop' testing was followed by flight tests with all the equipment onboard the aircraft turned on, and the aircraft autonomously flying around. However, excessive colored noise, as high as 40% noisy pixels of overall video stream, was observed in such images. Many of the images individually had even 90% of the image interfered by noise. This was further investigated as follows.

D. Noise Analysis and Handling

The excessive noise that was encountered, could have been due to the following: [13]

1) Interference with other avionics,
2) Noise due to wireless link,
3) Electronic noise in digital camera and circuitry
4) Shot noise of an ideal photon detector etc.

This noise was giving false positives and was generating higher confidence levels than an actual blue LED, for the chosen 5 parameters. In order to handle this noise, we initially considered various approaches as discussed in Section I-D. However, use of any such approaches would have increased the computational load even further. We were already restricted to half the block size as the grid step due to computational load as discussed (in Section IV-C) above. Hence a technique to handle this excessive noise was required that would be very low cost and yet very effective in a real-time scenario.

Upon further image analysis of various noisy images, which individually had up to 90% of the image distorted by noise, revealed the following (some noisy images are shown in Fig. 13, with representative pixels marked at left bottom),

1) The noise generated was either very low on all three channels (very low intensity),
2) Or, was very high on just one channel (out of red, blue or green),
3) Or was very high on any two of the three channels,
4) Almost no such noise was encountered that was high on all three channels (very high intensity).

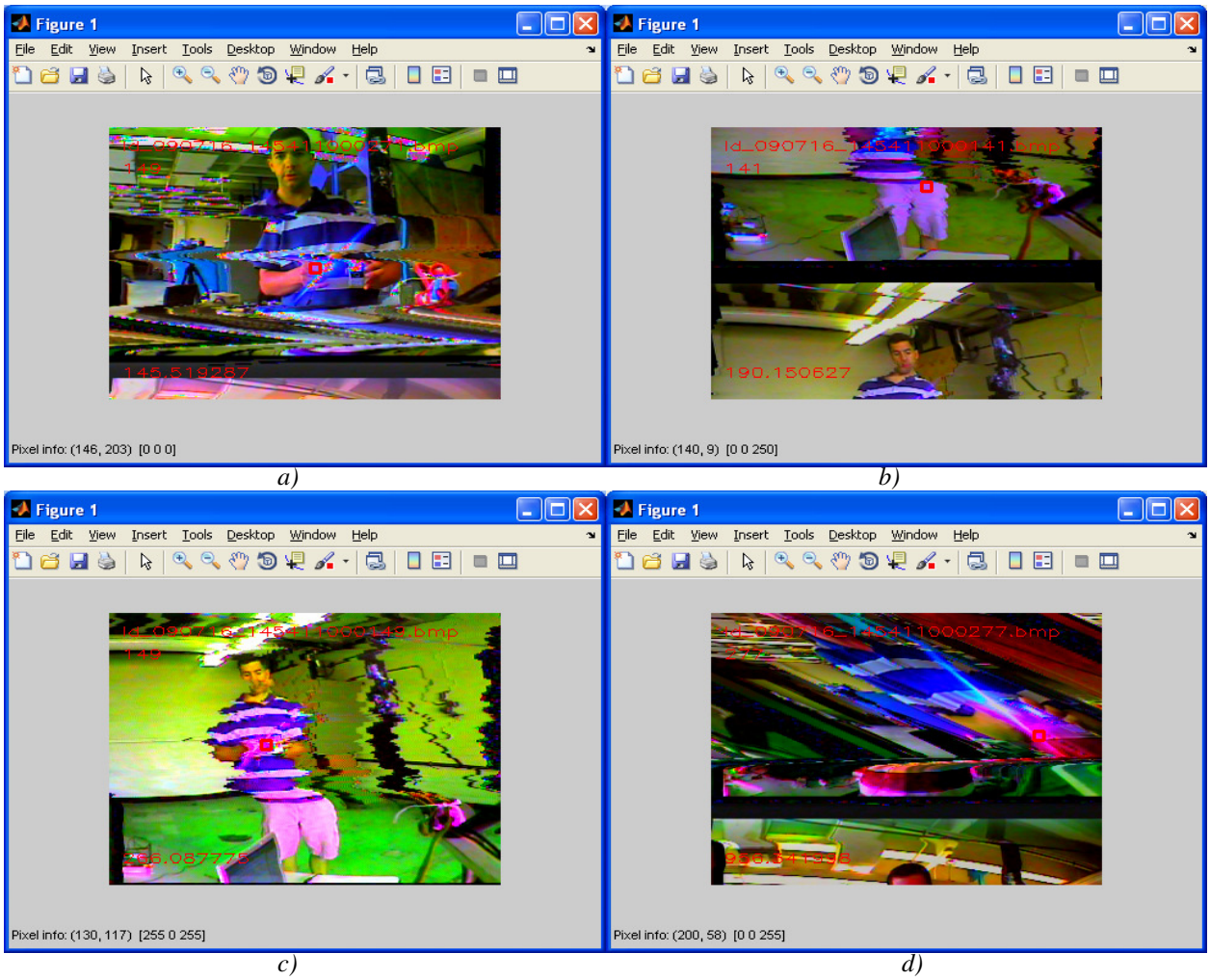


Figure 13. Noisy Images. *Up to 40% of our total digitized images in a video stream were noisy initially. On some of the individual images, noise was as high as 90-100%. Noisy pixels were either very low on all three channels (Fig. a) or were very high on one or two channels (Fig. b, c & d). Almost no noise was encountered that was high on all three channels. Cases b, c and d included blue channels and gave very high confidence levels (false positives) for our initial algorithm of blue LED detection.*

Hence, whenever this noise was very high on either blue channel alone, or any two of the three channels, one being blue, it gave pretty high confidence levels, much more than a blue LED. A blue LED therefore, could not be successfully detected in a given video stream in the presence of such noise.

This analysis of noise however suggested that if a minimum threshold is put on all three channels simultaneously, all noisy blocks of pixels could be avoided. Hence, a block of pixels if does not pass all these three thresholds simultaneously, should not be considered as a candidate for blue LED detection.

It may be realized that this thresholding for noise did not require any further calculations to the original proposed algorithm, as the three channels were to be calculated any way for every block for the linear confidence level calculation algorithm. Instead, implementing these thresholds for noise, in fact reduced the computational load, as further calculations and processing for such noisy block of pixels was not required if these did not pass all the three thresholds simultaneously. Subsequent flight tests revealed successful detection of a blue LED without having false positives from noisy images or noisy blocks of pixels.

E. Final Implemented Algorithm

Based on 'hardware in the loop tests, flight tests and noise analysis, it was decided to use a mix of both multiple thresholds and linear confidence level algorithms. Hence out of five thresholds as proposed in the multiple thresholding algorithm above (Section III-B), following minimum thresholds were chosen:

1) Min_B,
2) Min_I

Based on the noise analysis (Section IV-D above), two more thresholds were added as follows:

3) Min_R,
4) Min_G

After a block of pixels would pass all these four thresholds, linear confidence level calculations were done for all the five parameters, as described in Section III-C above. The final implemented algorithm is as depicted in Fig. 14.

Sample output images from this final algorithm, from HITL tests are presented in Fig. 15 and from actual flight tests are presented in Fig. 16.

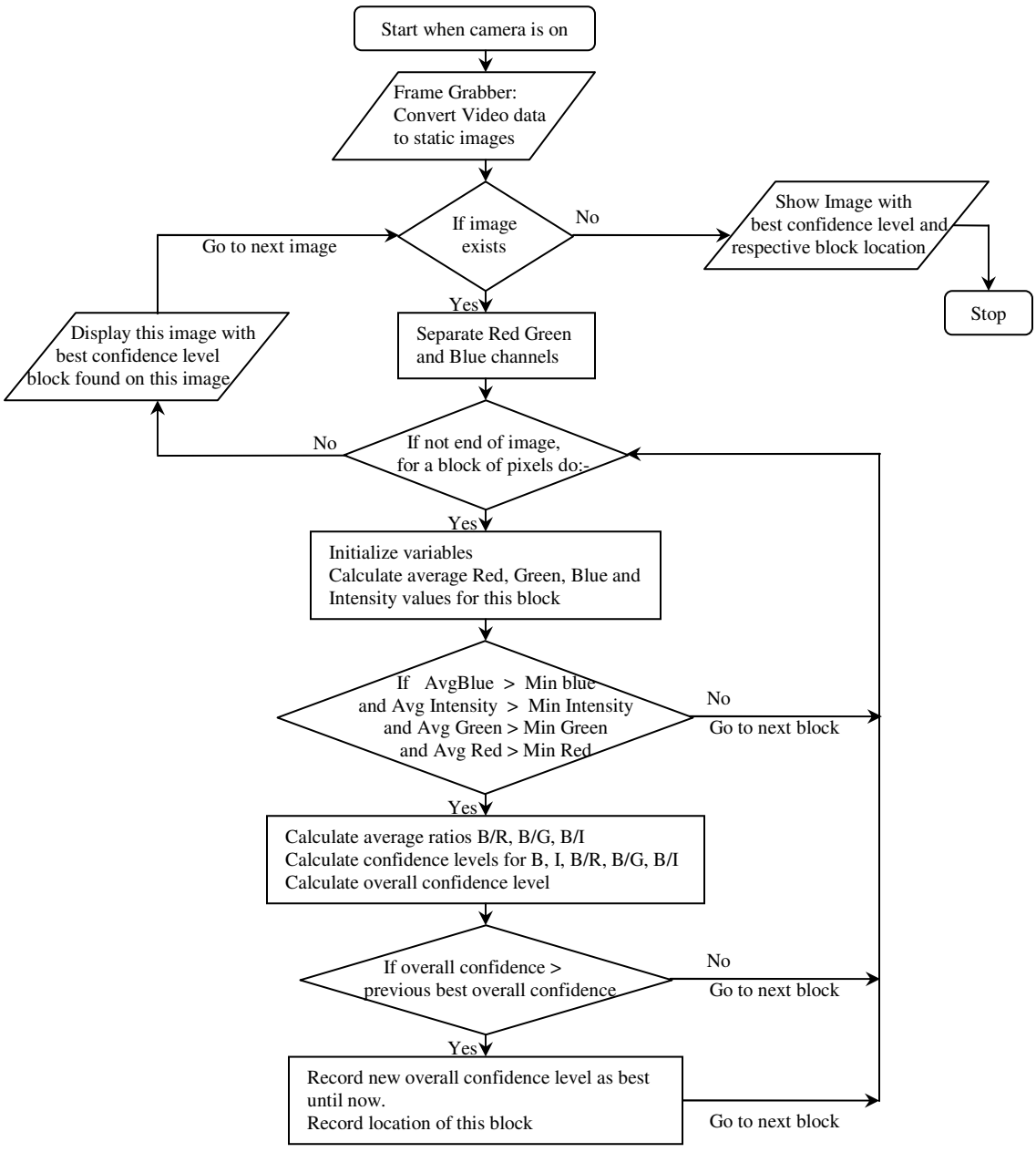


Figure 14. Final Implemented Algorithm

a) Best confidence level from 6 feet away *b) Best of whole video found at 5 feet away*

Figure 15. Output Images from HITL Tests. *The images depict successful detection of blue LED with highest confidence level at the location of the blue LED within an image (a), and picking the image with highest confidence level of all images in the whole video stream (b).*

a) Best confidence level from 5 feet away *b)Best of whole video found at 4 feet away*

Figure 16. Output Images from Flight Tests. *The images depict successful detection of blue LED with highest confidence level at the location of the blue LED within an image (a), and picking the image with highest confidence level of all images in the whole video stream (b).*

V. Conclusion

We have presented a vision-based algorithm to detect a blue LED for GeorgiaTech Aerial Robotics Lama - Aircraft in real time and in the presence of excessive color noise. Various image analysis techniques, including color histograms, filtering techniques and color space analyses were used to establish typical pixel-based characteristics of a blue LED in an image. A block based search algorithm was used to search for those established characteristics in real-time image data stream from a colored camera. A confidence level was defined based on all the significant criteria, so that higher the confidence level, more was the probability of having found the blue LED in the image stream. The algorithm was implemented on the GTAR lama aircraft and was later refined based on flight tests. A very low cost noise & interference filter was implemented, in order to handle excessive noise that was encountered during flight tests. This efficiently eliminated the need for processing the unwanted or noisy pixel blocks, thus reducing computational cost quite significantly. The final algorithm used both multiple thresholds as well as linear confidence level calculation in a block based search approach. This algorithm now is capable of

handling as high a noise and interference as 30-40% in the image data and has been successfully used in Aerial Robotics competition to detect a blue LED in real time.

References

[1] http://iarc.angel-strike.com/oldauvs/5th_mission/index.php#

[2] Celik,T; Kai-Kuang Ma; "Computer Vision Based Fire Detection in Color Images", *2008 IEEE Conference on Soft Computing in Industrial Applications, SMCia/08, 258-63, 2008.*

[3] Candamo,J; Kasturi,R;Goldgof,D; "Using Color Profiles For Street Detection in Low-Altitude UAV Video", *Proceedings of the SPIE – The International Society for Optical Engineering, v7307,p73070O(10pp.),2009.*

[4] Chung, Kuo-Liang; Lin, Yi-Ru; Huang, Yong-Huai; "Efficient Shadow Detection of Color Aerial Images Based on Successive Thresholding Scheme", *IEEE Transactions on Geoscience and Remote Sensing, v47, n2, p671-682, February 2009.*

[5] Juang, Chia-Feng; Sun,Wen-Kai, Chen, Guo-Cyuan, "Object detection by color histogram-based fuzzy classifier with support vector learning", *Neurocomputing, v72, n10-12, p2464-2476, June2009.*

[6] Li Da-Jie; Chai yi; Yin Hong-Peng; Xu Shu-Ping; "Method of Road Recognition Based on Color Image Edge Detection", *Computer Engineering and Applications, v44, n28, p177-9, 183, 1Oct. 2008.*

[7] Xu,S; "Robust Traffic Sign Shape Recognition Using Geometric Matching", *IET Intelligent Transport Systems, v3, n 1, p10-18, March 2009.*

[8] Huang, Shih-Yu; Tsai, Wei-Chang; "A Simple and Efficient Block Motion Estimation Algorithm Based on Full-Search Array Architecture", *Signal Processing Image Communication 19 (2004) 975-992*

[9] Rehan, Mohamed; El-Kharashi, M.W; Gebali, Fayez; "A Hierarchical Desing Methodology for Full-Search Block Matching Motion Estimation", *Multidim Syst Sign Process (2006) 17:327-341*

[10] Lim,Chan; Kang,Huun-Soo; Kim, Tae-Yong; Yoo,Kook-Yeol; "A Fast Full Search Algorithm For Variable Block-based Motion Estimation of H.264", *Advances in Visual Computing-First International Symposium, ISVC 2005 v 3804 LNCS, p 710-717*

[11] Dung,Dang; Luo,Wenbin; "Hybrid Vector Filtering for Noise Detection and Removal in color Images", *24th International Conference on Computers and Their Applications, CATA2009, p19-24,2009.*

[12] Said, Asaad F; "White and color Noise Cancellation Using Adaptive Feedback Cross-Coupled Line Enhancer Filter", *Proceedings of the 8th IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2008, p 207-212, 2008*

[13] Gonzalez, Rafael, C., Woods, Richard, E. and Eddins, Steven, L., *Digital Image Processing Using Matlab*, Pearson Prentice Hall, 2004.

[14] Gonzalez, Rafael, C., Woods, Richard, E., *Digital Image Processing*, 2nd Ed, Prentice Hall, 2002.

[15] Chowdhary, G., Ottander, J., Sala¨un, E., and Johnson, E., "Low Cost Guidance, Navigation, and Control Solutions for Miniature Air Vehicle in GPS Denied Environments", First Symposium on Indoor Flight, *International Aerial Robotics Competition, (2009).*